Table of Contents

|  |  |
| --- | --- |
| **Statement of Completeness** | **2 - 3** |
| **Unit Testing** | **3** |

Statement of Completeness

**Student Base Class:**

All functionality implemented correctly to the best of my knowledge.

Ran into problems when it came time to adding student objects to a heterogeneous list, and not understanding the data binding problems mentioned in the lectures correctly. However, making display a virtual function worked well.

I used enum types quite frequently, as I felt they made for a more elegant solution and obvious solution to containing Student Category types, then using a bool or string or something.

**Course Student Derived Class:**

All functionality implemented correctly as far as I know.

I decided to use a struct to contain the unit data (unit code and grades), and storing these struct objects in a vector.

**ResearchStudent Derived Class:**

All functionality implemented correctly as far as I know.

Again I was quite liberal with my use of enum types to represent both Degree Type and Study Status. I also use enum type VALUES as array indexes, which I feel is a more elegant solution then having a counter.

**Client Class:**

All functionality implemented correctly as far as I know.

I started this assignment a few days after it was released, and spent a LOT of time developing input validation methods that were so spectacularly modular and beautiful it made me cringe inside when I realised we are to assume all user input is correct.

I think developing a more generic displayStudents method, and then passing in the student type (course, research or all) would have been nicer than 3 separate ones.

**Unit Testing:**

Thoroughly confused as to what I was supposed to do. Was I meant to write test cases for every method? I see invalid input errors in the table in the document on blackboard, but I thought user input was assumed to be correct?

I did a few before giving up.

**Student Base Class Testing:**

|  |  |
| --- | --- |
| Category | Description |
| 1 | setFirstName(string firstNameInput) Check that getFirstName() returns the same. Last name testing is the same |
| 2 | setIdNumber(string idNumber) Check that getIdNumber() returns the same |
| 3 |  |
| 4 |  |
| 5 |  |
| 6 |  |
| 7 |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Category | Input | Expected Output | Actual Result | Pass/Fail |
| 1 | “Dominic” | “Dominic” | “Dominic” | Pass |
| 1 | “Dom-ini-c” | “Dom-ini-c” | “Dom-ini-c” | Pass |
| 1 | “Dom In Ic” | “Dom In Ic” | “Dom” | Fail |
| 2 | “n6869378” | “n6869378” | “n6869378” | Pass |
| 2 | “n 6869378” | “n 6869378” | “n” | Fail |
|  |  |  |  |  |
|  |  |  |  |  |